**Music Management Application**

### Document Properties

| Item | Details |
| --- | --- |
| Document | Architecture Design Document |
| Application | Music Management Application |
| Author | Prakash Reddy |
| Version | V1.0 |
| Creation Date | 09/02/2020 |
| Last Updated | 09/05/2020 |
| Status | Completed |

Contents

[Document Properties 1](#_Toc45989188)

[**1.** **Introduction** 3](#_Toc45989189)

[**2.** **Architecture Diagram** 3](#_Toc45989190)

[**3.** **Use-Case View** 4](#_Toc45989191)

[**4.** **Flow Chart** 5](#_Toc45989192)

[**5.** **Components** 6](#_Toc45989193)

[**6.** **Services** 6](#_Toc45989194)

[**7.** **Validations** 6](#_Toc45989195)

[**8.** **Tools and Libraries** 7](#_Toc45989196)

[**9.** **Unit Testing** 7](#_Toc45989197)

[**10.** **Responsive** 8](#_Toc45989198)

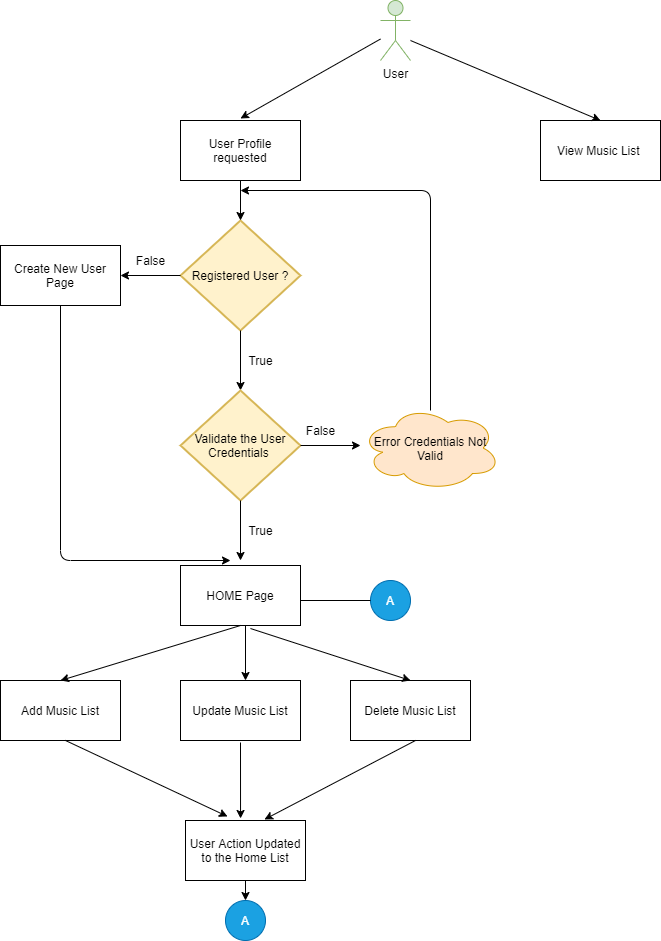
[**11.** **Accessibility** 8](#_Toc45989199)

[**12.** **Setup** 9](#_Toc45989201)

# **Introduction**

This document provides a comprehensive architectural overview of the application, using number of different architectural views to depict different aspects of the application.

# **Architecture Diagram**



# **Use-Case View**

The application use cases are -

* **Add Music List** – It provide functionality to add List, also having unique name validation with it.
* **Add Music to the List** – After adding list, card can be added inside a list. So, there will be a link to add a card with name and respective list. The card name also having unique name validation.
* **Edit List –** Any list can be editable while clicking on list edit button.
* **Delete List** – Each list will have a delete button, the list can be deleted by clicking on it, it also confirms about the deletion of list

These use cases are initiated by the user.

# **Flow Chart**

Music List View

Add Music

|  |
| --- |
| Home Page |

Add Music List

# **Components**

* AppComponent – This component is used for App Application
* HeaderComponent – This component is Sinup and Login& LogOut
* HomeComponent – This Component is List of Musics.
* List-Music Component – Home Page of application that show all the list and their respective cards with edit and delete feature and links for adding new list and cards.
* Add-Music ListComponent – This component contain form for add list.
* Edit-music ListComponent – This component contain form for edit list.

# **Services**

There are three services –

1. AuthService – This service generally having logic about list in the application and this contain code for getting only list and list with their cards, edit list details, add new list and delete list logic and login,SinUp,LogOut.
2. Theme-color Service– This Serve having a light theme and dark theme logic

# **Validations**

1) Application covered the required validation for login and Sign Up.

**Validation strategy:**

- Reactive form for validation

- There is corresponding html template for form elements

- Required attribute true for individual form element, this can be extended to have in a separate service file for re-usability.

# **Tools and Libraries**

- Angular version: 10.0.11

- TypesScript: 3.9.5

- Rxjs: 6.5.5

- Jest: 26.4.2

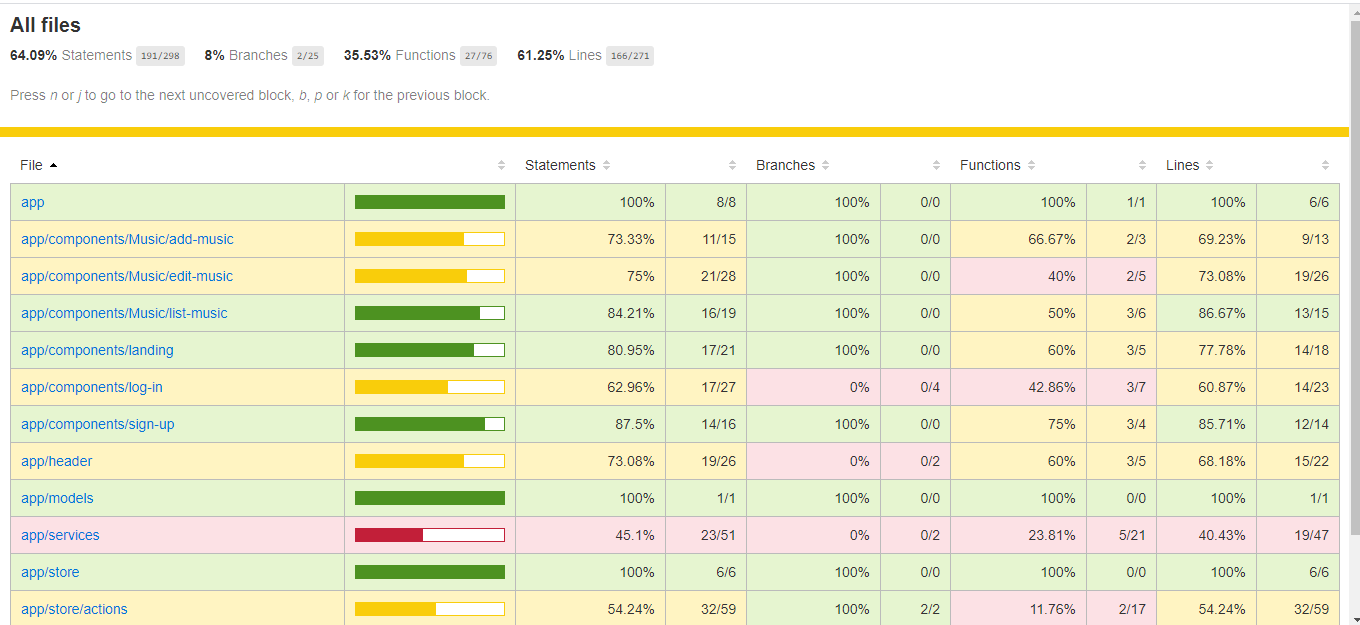
This is a unit testing framework; it consists assertion and mocking support for writing test cases.

# **Unit Testing**

Added Unit Test cases for services.

It consists of two services -

* + List.service.ts – It contain test cases for the all the logic like add list, update list, delete list and get list.
    - No of test cases: 13.



# **Responsive**

* The application is responsive with all the device and desktop size.
* Used CSS Grid and CSS for templating and Layout.

# **Accessibility**

For Accessibility, application having appropriate HTML tags that are preferred for accessibility

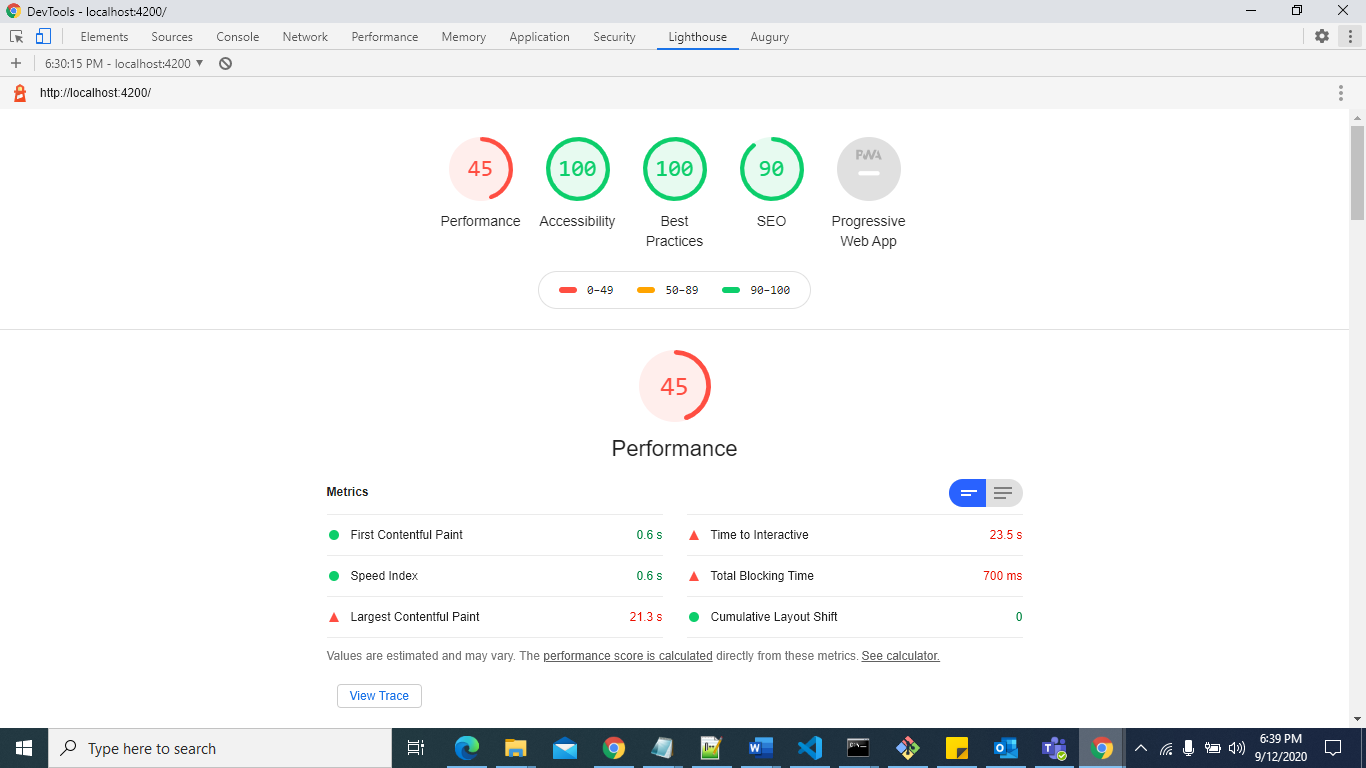
User can run application via keyboard and tab.

# **12.** **Storybook**

* I have installed and added basic setup for Storybook.
* All components rendering fine with default views

# **13.Lighthouse**

Mobile View



# **14.Setup**

1. Download and install node.js (version: 12.16)
2. Install npm and install @angular (version: 9.0)
3. Git clone <https://github.com/arpitaPersistent/Sample-Application.git> origin master
4. Navigate to cloned folder & run npm install
5. To run API(JSON Server): npm run json:server
6. To run application: ng serve
7. For testing : npm run test-watch
8. For Storybook: npm run storybook
9. Login

Email: [prakash0544@gmail.com](mailto:prakash0544@gmail.com)

Password: test